

Blue Gene/P Architecture

Vitali Morozov

Performance Engineering Group

Argonne Leadership Computing Facility

Fall Getting Started Workshop

October 4, 2011, Argonne National Laboratory

Overview

- Blue Gene/P – general overview
- ASIC – the core of the system
- PowerPC 450 processor details
- Single core performance
- Cache and memory hierarchy
- Performance counters
- DMA engine
- Interconnects and performance
- Programming models
- Development environment



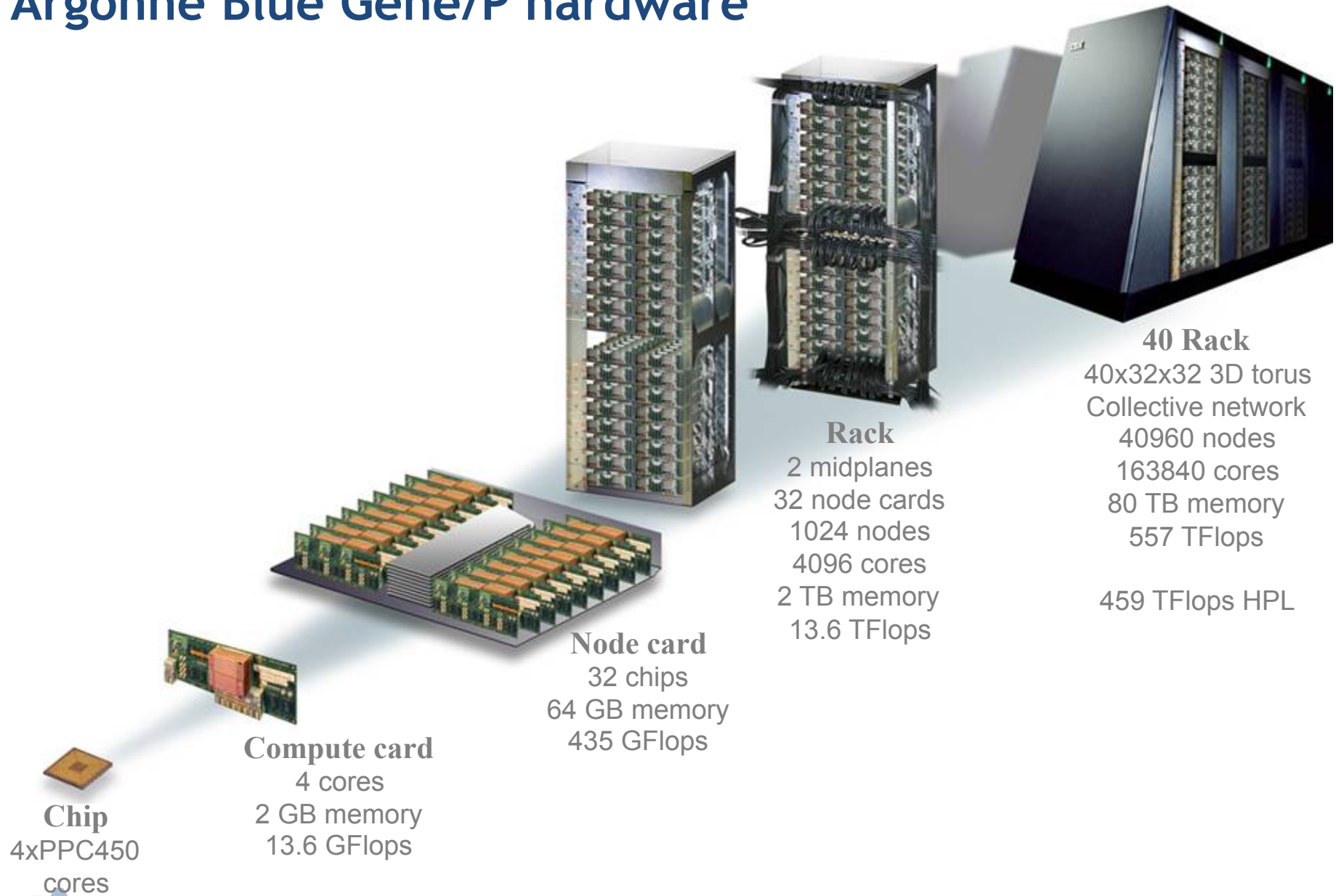
Blue Gene/P: A summary of facts

- June 2007, availability is announced by IBM
- November 2007, Argonne announces BG/P 40-rack acquisition
- June 2008, Argonne #1 for Open Science, #3 in Top500
- November 2010, Argonne is #13 in Top500
 - A midplane BG/Q prototype I was demonstrated, #115, 65.35 TF / 104.86 TF peak
- June 2011, Argonne is #15 in Top500
 - A midplane BG/Q prototype II was demonstrated, #109, 85.88 TF / 104.86 TF peak

TOP 500 List	# places	# racks	Performance	Argonne
June 2008	12	91 racks	1.24 PF	#3
November 2010	13	217 racks	2.84 PF	#13
June 2011	10	190 racks	2.58 PF	#15

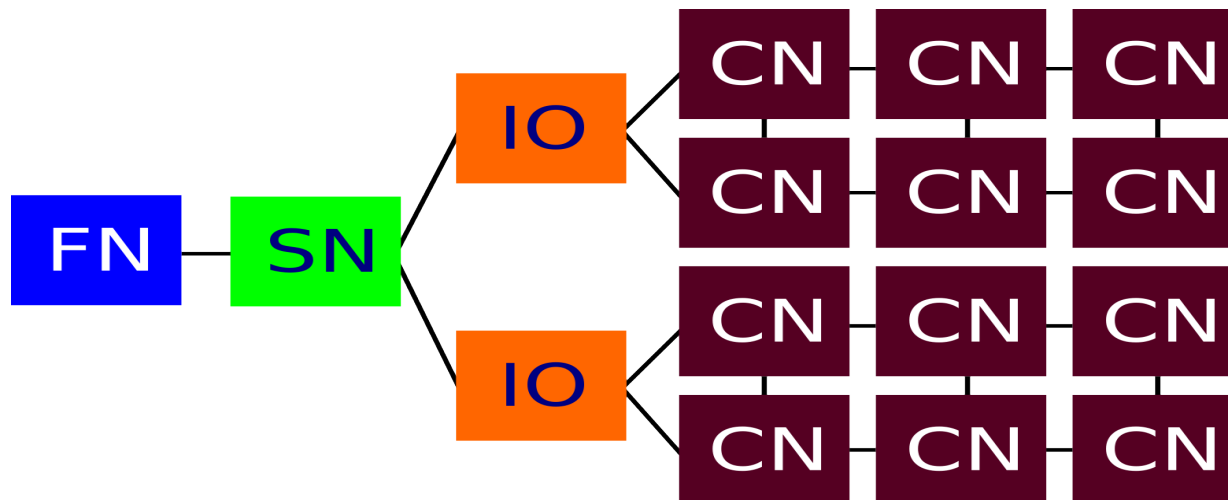


Argonne Blue Gene/P hardware

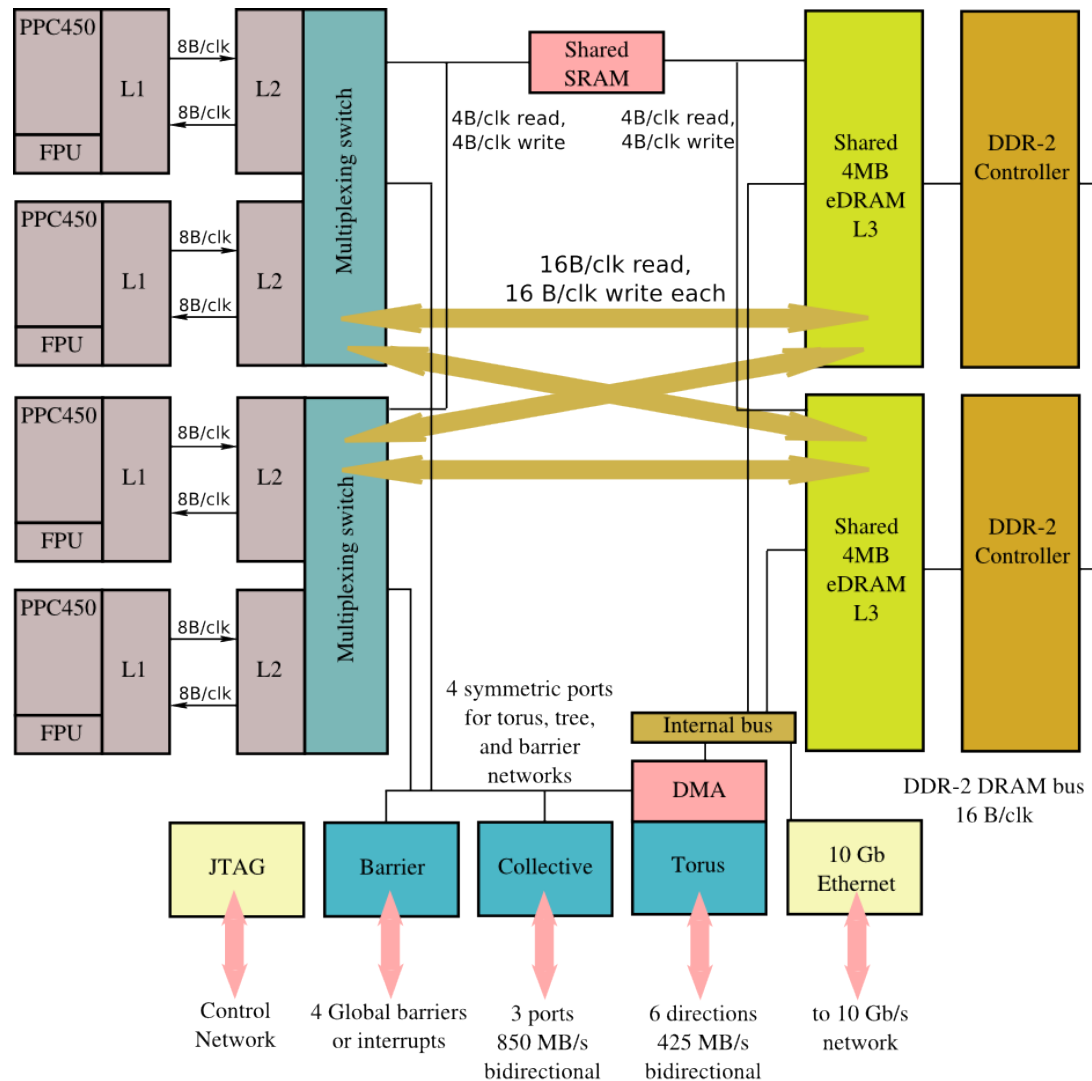


Blue Gene Hierarchical Organization

- **Front-end** nodes - dedicated for users to login, compile programs, submit jobs, query job status, debug applications
- **Compute nodes** – run user applications, use simple compute node kernel (CNK) operating system, ship I/O-related system calls to I/O nodes
- **I/O nodes** – provide a number of Linux/Unix typical services, such as files, sockets, process launching, signals, debugging; run Linux
- **Service nodes** - perform partitioning, monitoring, synchronization and other system management services. Users do not run on service nodes directly.



Application Specific Integrated Circuit



Three modes of execution

■ SMP mode

- `qsub --mode smp`
- Single MPI task on CPU0 / 2 GB RAM

■ Dual mode

- `qsub --mode dual`
- Two MPI tasks on a node / 1GB RAM each

■ Virtual Node mode

- `qsub --mode vn`
- Four MPI tasks on a node / 512 MB RAM each



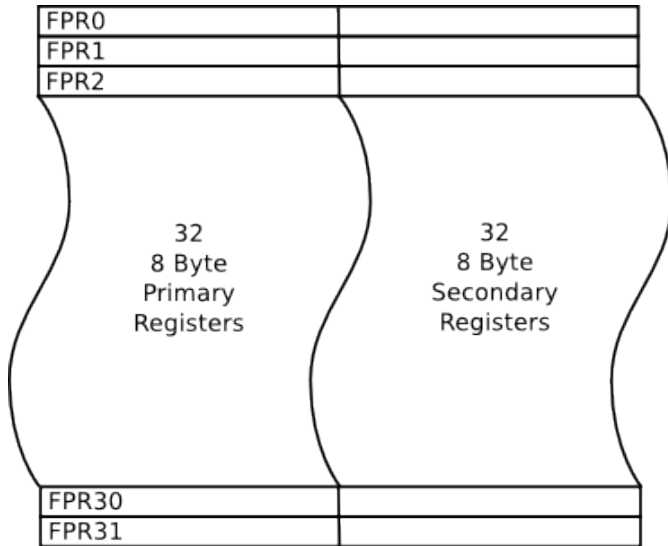
PowerPC 450 Processor

- A brunch of PowerPC440 Processor
- Dual-issue single-threaded embedded 32 bit processor @ 850 MHz
- Two integer units, single load/store unit, special double FPU
- Three execution pipes and a two-way F-pipe
 - complex integer I-pipe for arithmetic, logic, and system management
 - simple integer J-pipe for arithmetic and logic instructions
 - L-pipe for loads, stores, and cache management
- Double FPU supports
 - standard FP PowerPC instructions on fpu0
 - SIMD instructions for 64-bit fp-numbers (fpadd, fpmul, fpmadd, fpre, ...)
 - 5 cycles fp pipeline
- L1 cache: 32iKB+32dKB, 32 Byte line size, 4 core's coherent
- L2 cache: prefetch buffer with 16 128-byte lines (2KB)



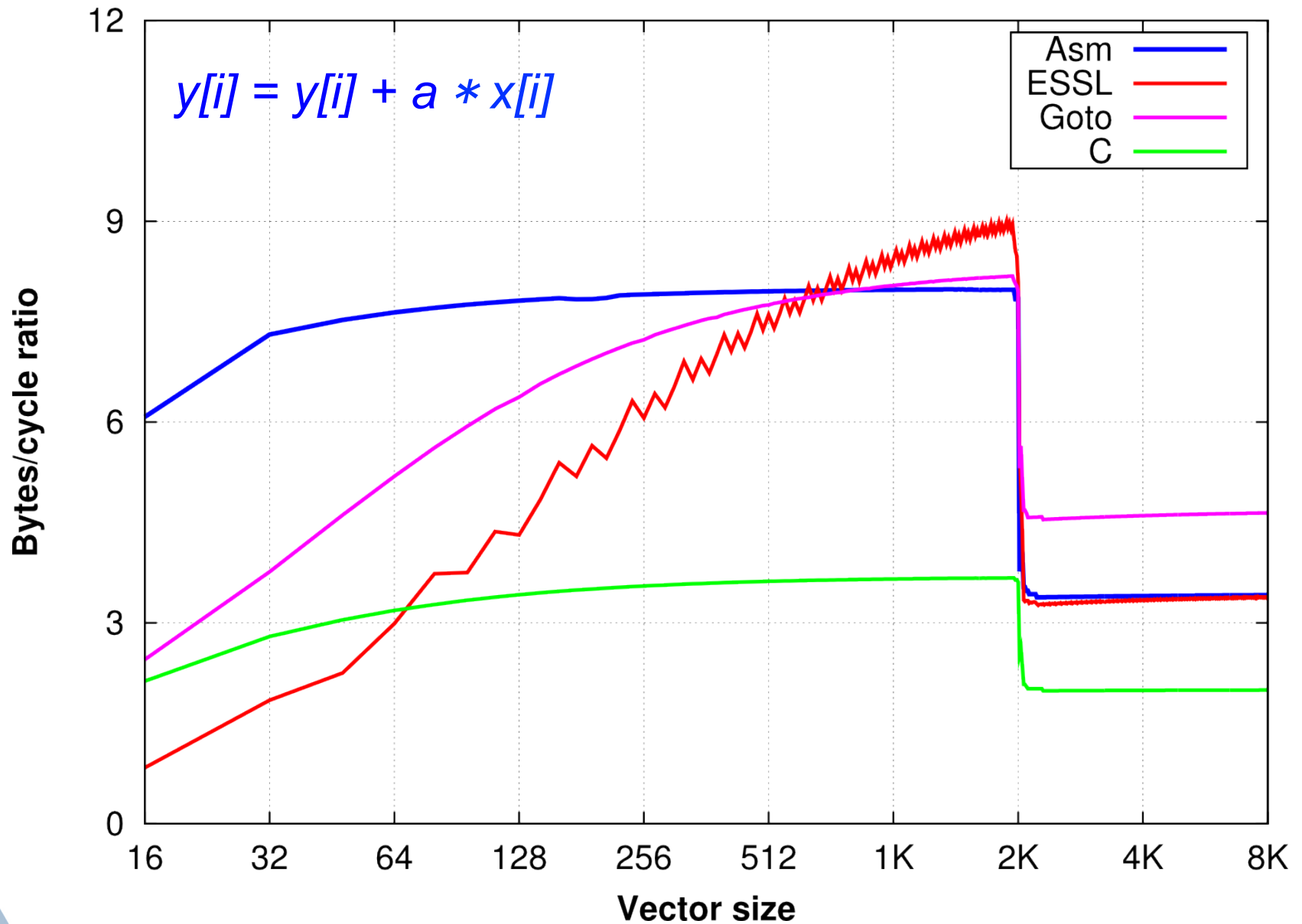
Double FPU Unit

fpu0 fpu1
primary secondary



- 2 64bit FP registers, primary and secondary
- Primary registers are “regular” FPU registers
- SIMD-type, both registers perform the same instruction
- XL compiler can generate double FPU code in limited cases
- Extensive set of parallel and cross SIMD instructions
- Instructions available as intrinsic functions
- Quad word loads/stores require 16-Byte alignment
- Hand-tuning for critical sections of the code

BG/P DAXPY Performance



BG/P DGEMM Performance

$C = A * B$ for square matrices

per-core performance (in % of peak)

N	Goto BLAS 1 thread	ESSL 1 thread	ESSL-smp 4 threads
32	13.96	10.68	11.79
100	54.96	44.48	2.79
128	61.82	55.81	5.50
200	64.50	64.89	16.30
256	71.18	63.19	27.33
500	74.01	75.04	59.71
512	74.65	72.89	58.18
1000	76.56	77.88	78.88
1024	76.62	75.59	75.87
4096	77.96	78.75	82.56
5000	78.04	81.28	85.60
7500	78.09	81.59	86.85



L1 Cache

Architecture

- *32 KB Instruction cache*
- *32 KB Data cache*
- *each private for each core*
- *each 32 Byte lines, 64 way-set-associative, 16 sets*
- *round-robin replacement*
- *write-through mode*
- *no write allocation*

Performance

- *at L1 load hit*
8 Bytes/cycle with 4 cycle latency
- *at L1 load miss, L2 hit (stream)*
4.6 Bytes/cycle, 12 cycle latency
- *at store:*
limited by external logic to one request every 2.9 cycles
about 5.5 Bytes/cycle peak



L2 Cache and Prefetch Unit

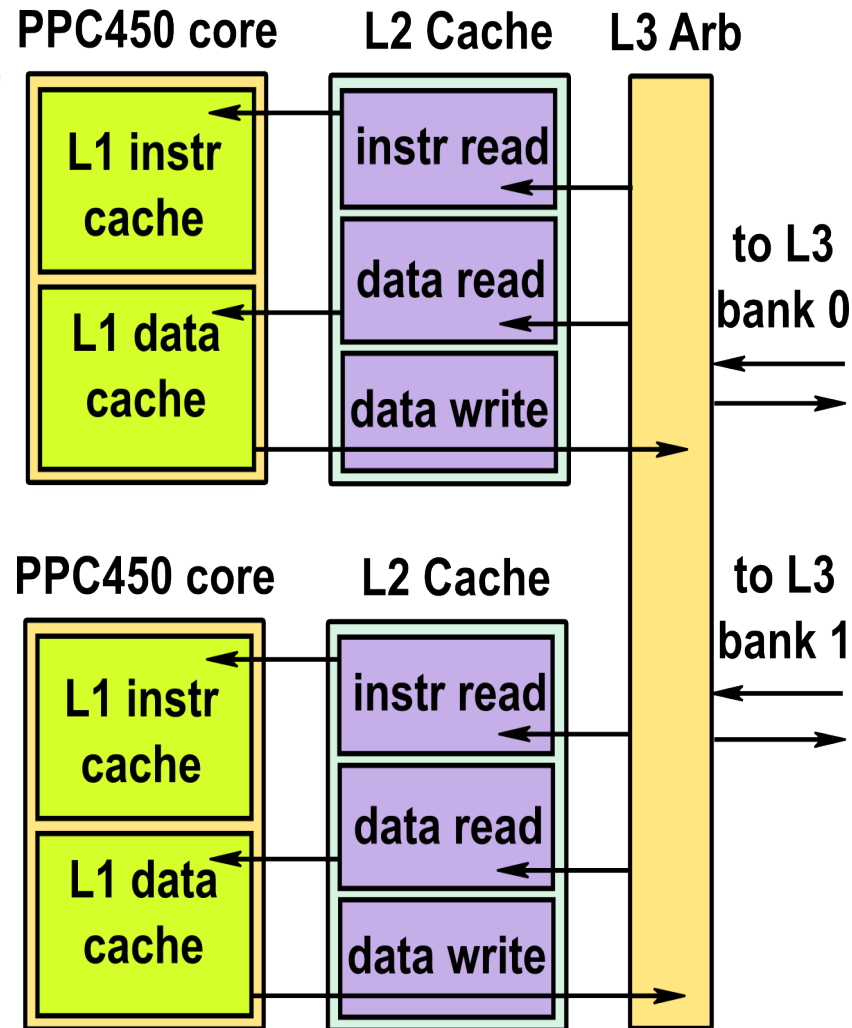
Contains three independent ports:
instructions read, data read, and data write

Performs two functions:
L3 arbiter and Prefetch engine

L3 arbiter
switch between L3 banks
can issue 1 read and 1 write to each bank
every 1/2 clock cycles (425 MHz)

Prefetch engine
15 entry fully associative prefetch buffer
each entry is 128 Byte line
can perform 1 or 2 line deep prefetching

Performance
12 cycle latency, 4.6 Bytes/cycle
up to 7 streams are supported





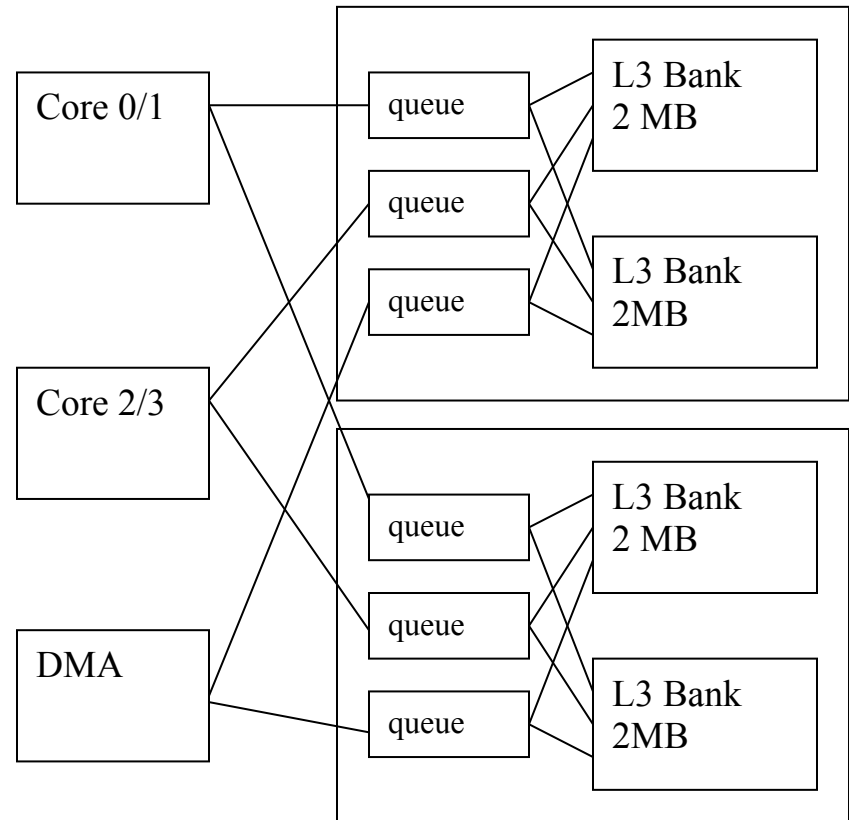
L3 Cache

4 banks of embedded DRAM
per node (8MB total)

Each bank contains

an L3 directory

15 entry 128B-wide
write combining buffer



Memory Subsystem

2 memory controllers are on the chip

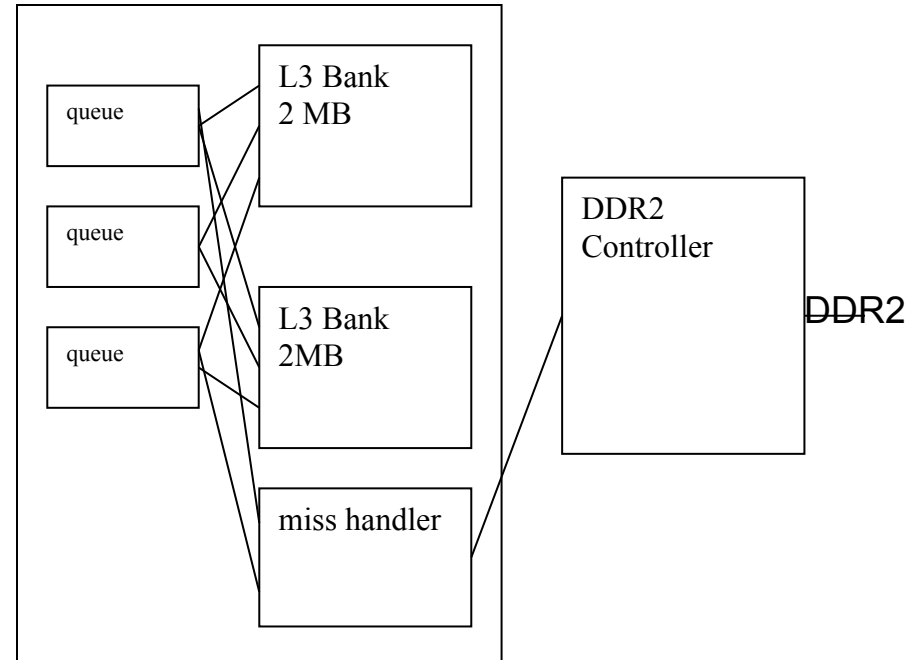
20 8b-wide DDR2 modules per controller fused on compute card

L2 – L3 switch is not a full core to L3 bank crossbar

Request rate and bandwidth are limited if two cores of one dual processor group access the same L3 cache bank

DDR2 is having 4 banks on 512Mb DDR module (2GB/node total). The banks support command reordering based on bank availability

Peak bandwidth only achievable if accessing 3 other banks before accessing the same bank again



Memory Hierarchy in Numbers

■ L1 Instruction and L1 Data caches

- 32 KB total size, 32-Byte line size, 64-way associative, round-robin
- `-qcache=level=1:type=d:assoc=64:line=32:size=32:\`
`level=1:type=i:assoc=64:line=32:size=32`

■ L2 Data cache

- 2KB prefetch buffer, 16 lines, 128-byte a line
- `-qcache=level=2:type=c:line=128:size=2`

■ L3 Data cache

- 8 MB, 35 (53) cycles latency
- `-qcache=level=3:type=c:line=128:size=8192:cost=35`

■ Memory size

- 2GB DDR-2 at 400 MHz, 86 (138) cycles

■ Memory bandwidth

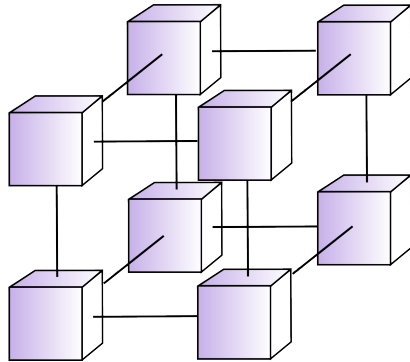
- in L1-cache: `lfpdx/stfpdx` instructions, 1 quadword load/cycle: $16\text{B} \times 850\text{ MHz} = 13.6\text{ GB/s}$
- out of L1-cache: complex memory hierarchy

Performance Counters Hardware

- Single Universal Performance Counter (UPC) unit per node
- Status register
 - Selects the mode for all counters (0/1 or 2/3)
 - Contains control bit to start/stop counting
- Configuration registers
 - 32 registers, 1 per 8 counters, 4 bits per counter
 - Selects: falling edge, rising edge, level high, level low
 - Defines the mode (0 or 1 if 0/1, 2 or 3 if 2/3)
 - Enables threshold interrupt
- Threshold register
 - Sets count value to enable interrupt
- 256 64 bit RW counter registers
 - Memory mapped
 - Count 1024 signals

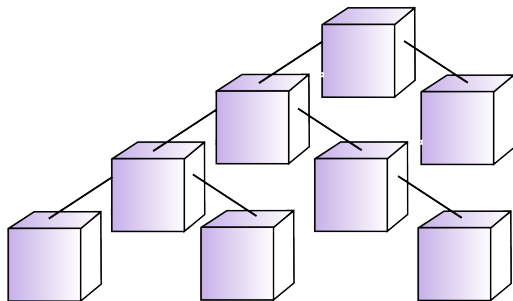


Interconnect Networks



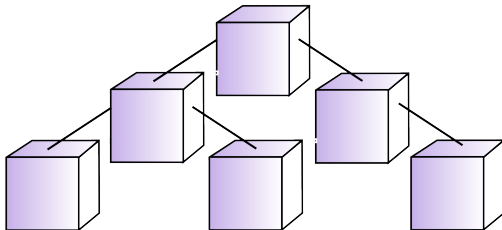
3 Dimensional Torus

- Basis for point-to-point communications
- Connects all compute nodes
- Supports virtual cut-through hardware routing
- 3.4 Gb/s on all 12 links (5.1GB/s per node)
- Hardware latency: 0.5 μ s per hop, 5 μ s farthest link
- MPI latency: 3 μ s per hop, 10 μ s farthest link



Collective Network

- Basis for collective and I/O communications
- Connects all compute and I/O nodes
- Supports integer and double reductions
- 6.8 Gb/s of bandwidth per link per direction
- Hardware latency: 1.3 μ s per tree traversal
- MPI latency: 5 μ s per tree traversal



Global Barrier and Interrupt Network

- Hardware latency: 0.65 μ s
- MPI latency: 1.6 μ s

DMA Hardware

Designed to offload the cores from data movement

Injects / receives data to and from **torus network**

Supports memcpy operations within the node

Has direct access to L3 cache

Uses physical addresses

Supports programming at SPI level

Supports arbitrary offsets

Message types:

- Memory Fifo – operates by memory located Fifos
- Direct Put – copies data directly to a remote destination address
- Remote Get – places data on remote node from given address

Basic Constructs

Injection and Reception Fifos

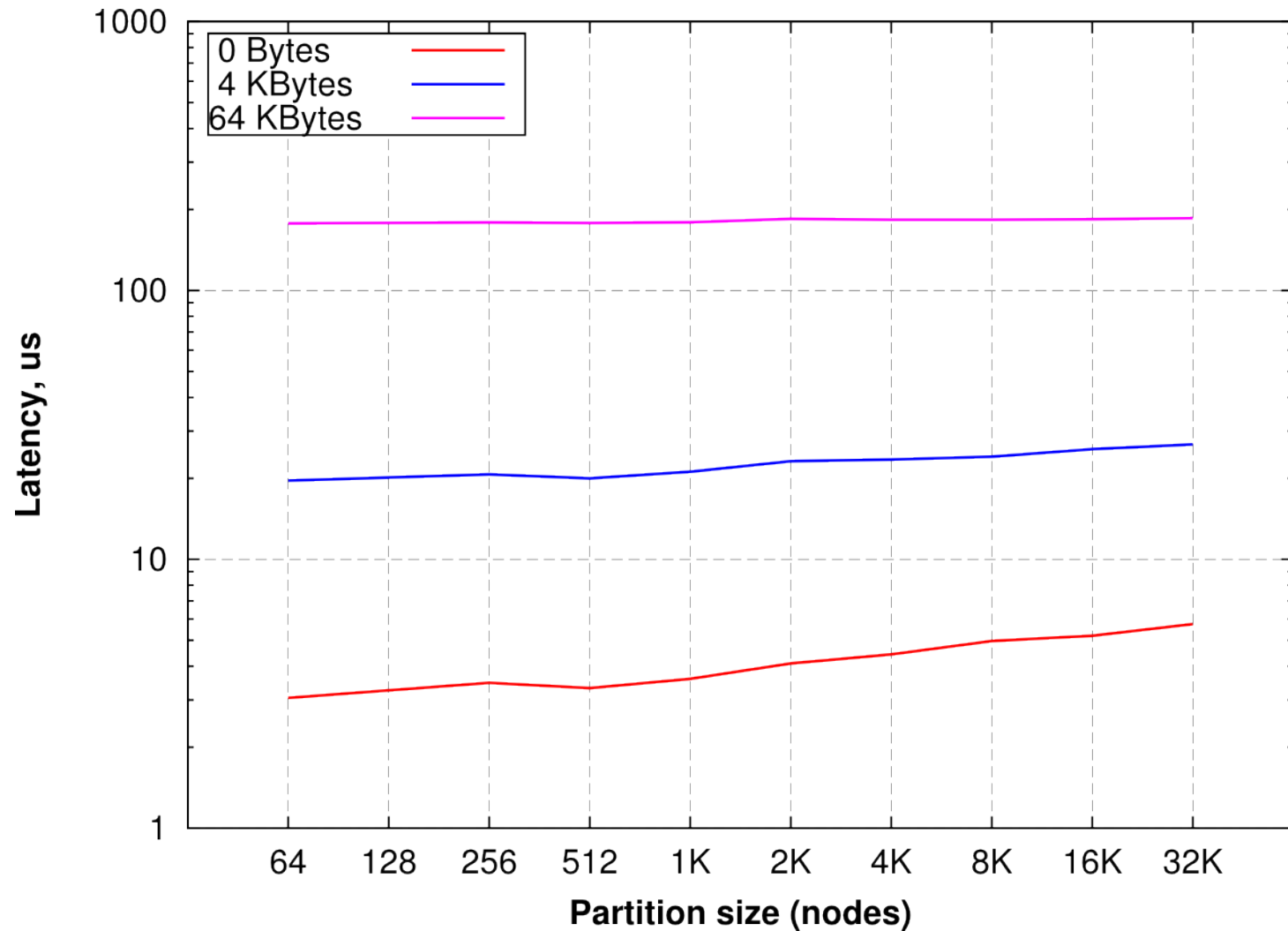
Message Descriptors

32B structures (destination, message type, message length,
payload pointer)

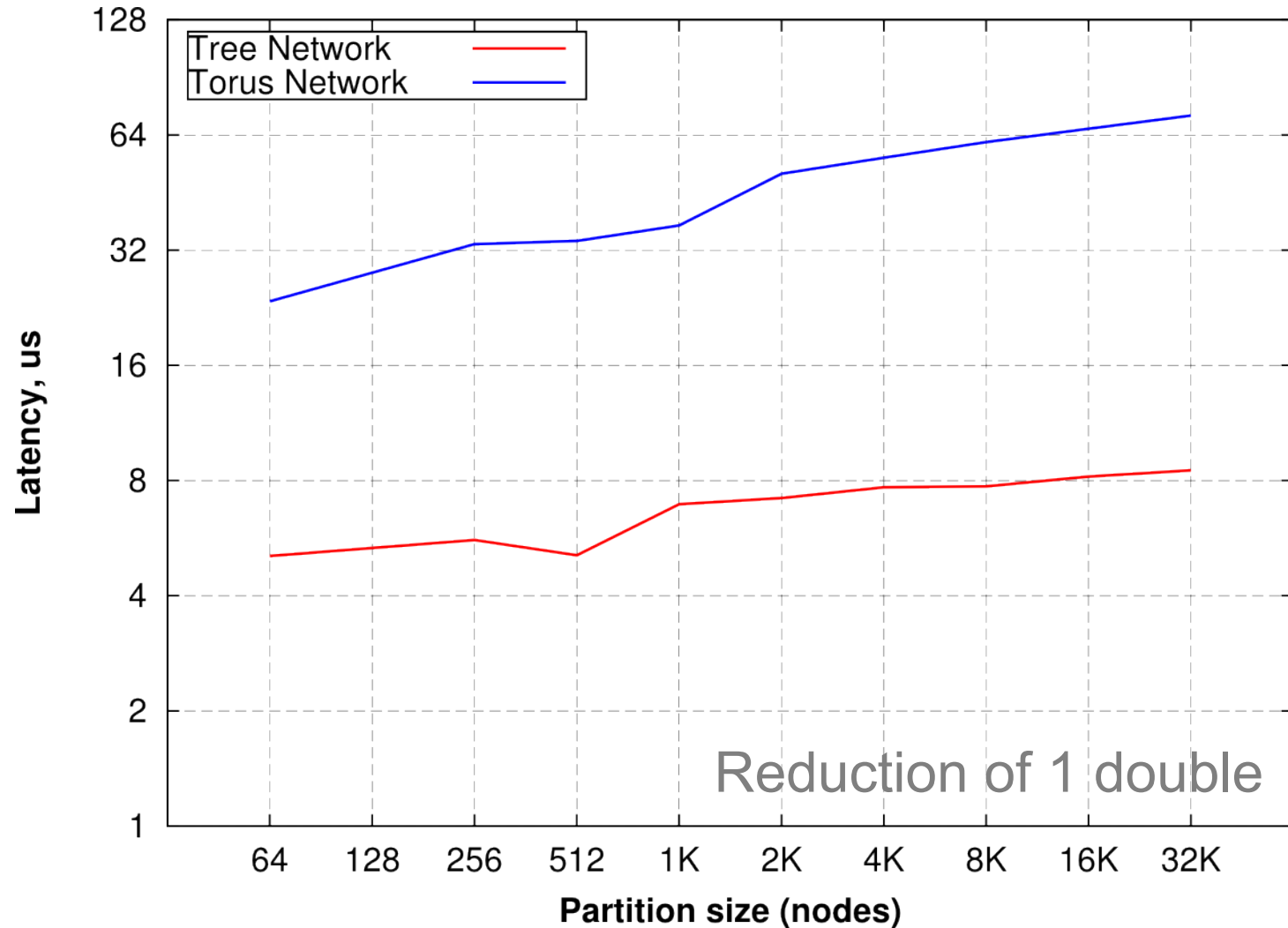


Torus interconnect performance

ping-pong latency between the farthest nodes



Collective interconnect performance



Blue Gene/P Input/Output hardware

- Scalable configurations with different compute to IO node ratio
 - 16, 32, **64**, or 128 to 1
- I/O node
 - Max bandwidth per IO Node is 1250 MB/s
 - Limited by the 10 Gb/s Ethernet link
- Streaming I/O performance
 - Can scale linearly if parallel IO is used

File size	1 rack	4 racks
16 GB	2148 MB/s	3674 MB/s
64 GB	2359 MB/s	4632 MB/s
128 GB	2354 MB/s	5042 MB/s

- Supports intensive tuning
 - requires sys-admin privileges, should work with us, if necessary

Programming Models and Development Environment

■ Familiar methods

- SPMD model - Fortran, C, C++, Python with MPI (MPI1 + subset of MPI2)
 - *Full language support with IBM XL and GNU compilers*
 - *Automatic SIMD FPU exploitation (limited)*
- Linux development environment
 - *User interacts with system through front-end nodes running Linux – compilation, job submission, debugging*
 - *Compute Node Kernel provides look and feel of a Linux environment*
 - POSIX routines (with some restrictions: no fork() or system())
 - pthread support, additional socket support
 - *Tools – support for debuggers, MPI tracer, profiler, hardware performance monitors, visualizer (HPC Toolkit), PAPI*

■ Restrictions (which lead to significant benefits)

- *Space sharing - one parallel job per partition of machine*
- *Virtual memory is constrained to physical memory size*



Resources

- ALCF Resource page

<http://www.alcf.anl.gov/support/usingALCF/index.php>

- ALCF FAQ Wiki Page

<https://wiki.alcf.anl.gov/index.php/FAQ>

- ALCF Getting Started Documentation

<http://www.alcf.anl.gov/support/gettingstarted/index.php>

- ALCF E-mail Support Line

support@alcf.anl.gov

- IBM RedBooks:

Compiler User Guides, Application Development Manuals

<http://www.redbooks.ibm.com/redbooks.nsf/redbooks/>

